Universität Zürich[UZH]

Institut für Informatik

Informatik II: Modellierung
Prof. Dr. Martin Glinz

Kapitel 6

Funktionsmodellierung II: Datenflussmodelle

© 2008-2016 Martin Glinz. Alle Rechte vorbehalten. Speicherung und Wiedergabe für den persönlichen, nicht kommerziellen Gebrauch gestattet; bei auszugsweiser Verwendung mit Quellenangabe. Verwendung für Unterrichtszwecke oder kommerziellen Gebrauch nur mit vorheriger schriftlicher Genehmigung des Autors.

Inhalt

6.1 Grundlagen

6.2 Datenflussdiagramme

6.3 Strukturierte Analyse

6.4 Methodik der Modellerstellung

6.5 Weitere Ansätze

6.1 Grundlagen

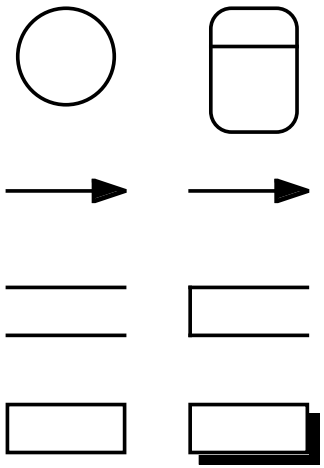
Datenflussmodelle beschreiben die Funktionalität eines Systems durch Aktivitäten und die Datenflüsse zwischen diesen Aktivitäten.

- Basis: Konzept der datengesteuerten Verarbeitung:
  - Eine Aktivität arbeitet, wenn ihre benötigten Datenflüsse eintreffen
  - Sie erzeugt bei ihrer Arbeit neue Datenflüsse
  - Diese steuern entweder andere Aktivitäten an oder verlassen das System als Ergebnis
- Der Steuerfluss ist implizit und wird nicht modelliert
- Ein reines Datenflussmodell modelliert nur Datentransport und Datenbearbeitung, nicht aber Datenspeicherung
- In der Praxis hinderliche Einschränkung ➪ Hinzufügen von Mitteln zur Modellierung von Datenspeicherung
- Bekanntester Vertreter: Datenflussdiagramm

6.2 Datenflussdiagramme

Ein **Datenflussdiagramm** (dataflow diagram, DFD) modelliert den Transport, die Bearbeitung und die Speicherung von Daten.

Zwei Notationen sind gebräuchlich:



Aktivität (Prozess; activity, process)

Datenfluss (dataflow)

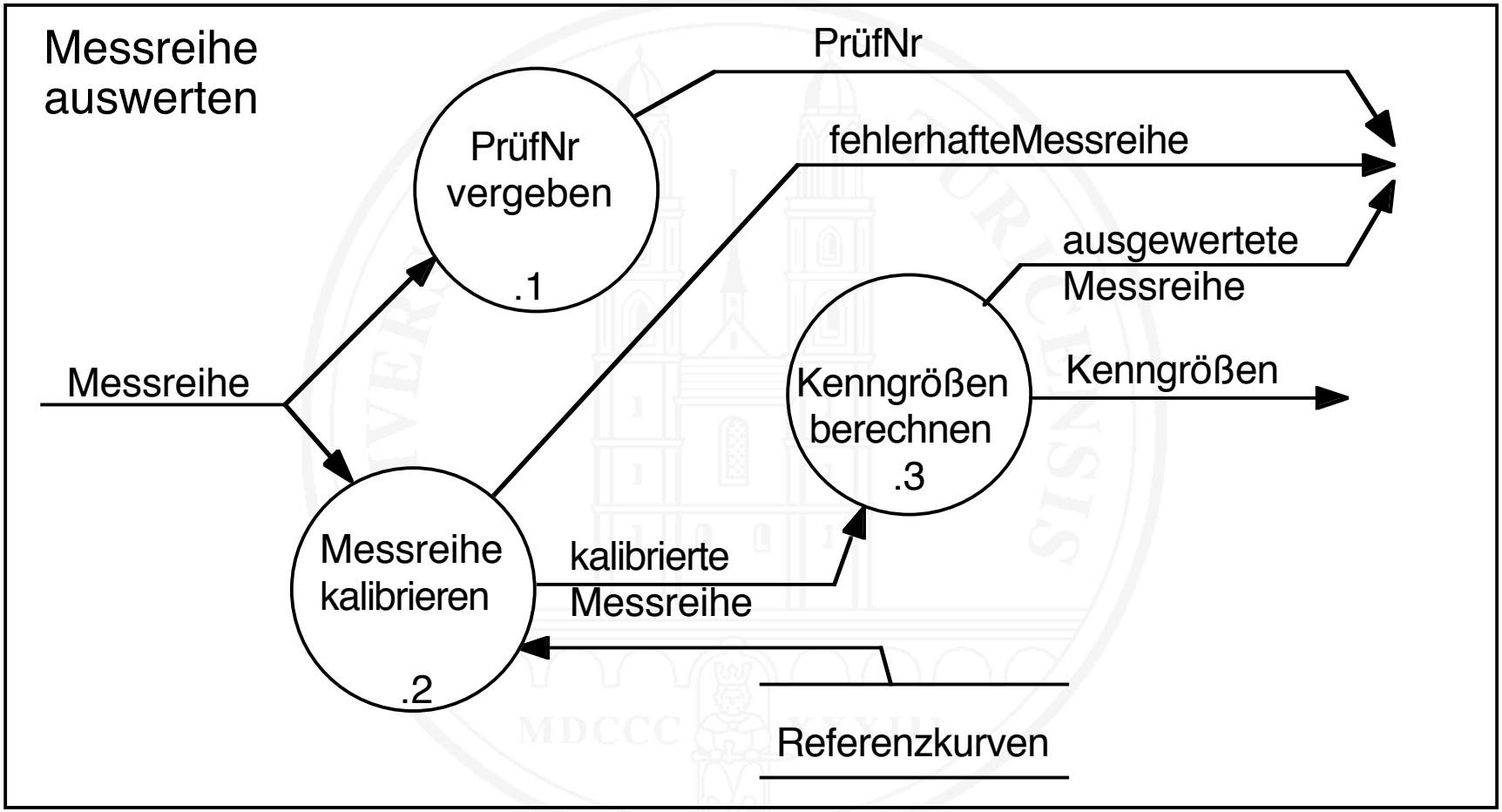
Speicher (store, file)

Endknoten (terminator, terminal, external)

Interpretation von Datenflussdiagrammen

- **Datenflüsse** transportieren Daten*pakete*, die von Aktivitäten oder Endknoten produziert bzw. konsumiert werden.
- **Aktivitäten** arbeiten nur dann, wenn alle von ihnen benötigten Eingabe-Datenflüsse vorliegen. Die Aktivität konsumiert die Daten, bearbeitet sie und produziert Ausgabe-Datenflüsse. Sie kann dabei zusätzlich Speicherinhalte lesen oder schreiben.
- **Speicher** modellieren Datenbehälter. Ihr Inhalt kann gelesen werden (ohne den Speicher zu verändern) und geschrieben werden (dabei wird der alte Inhalt zerstört).
- **Endknoten** sind Aktivitäten in der Systemumgebung.

Beispiel eines Datenflussdiagramms

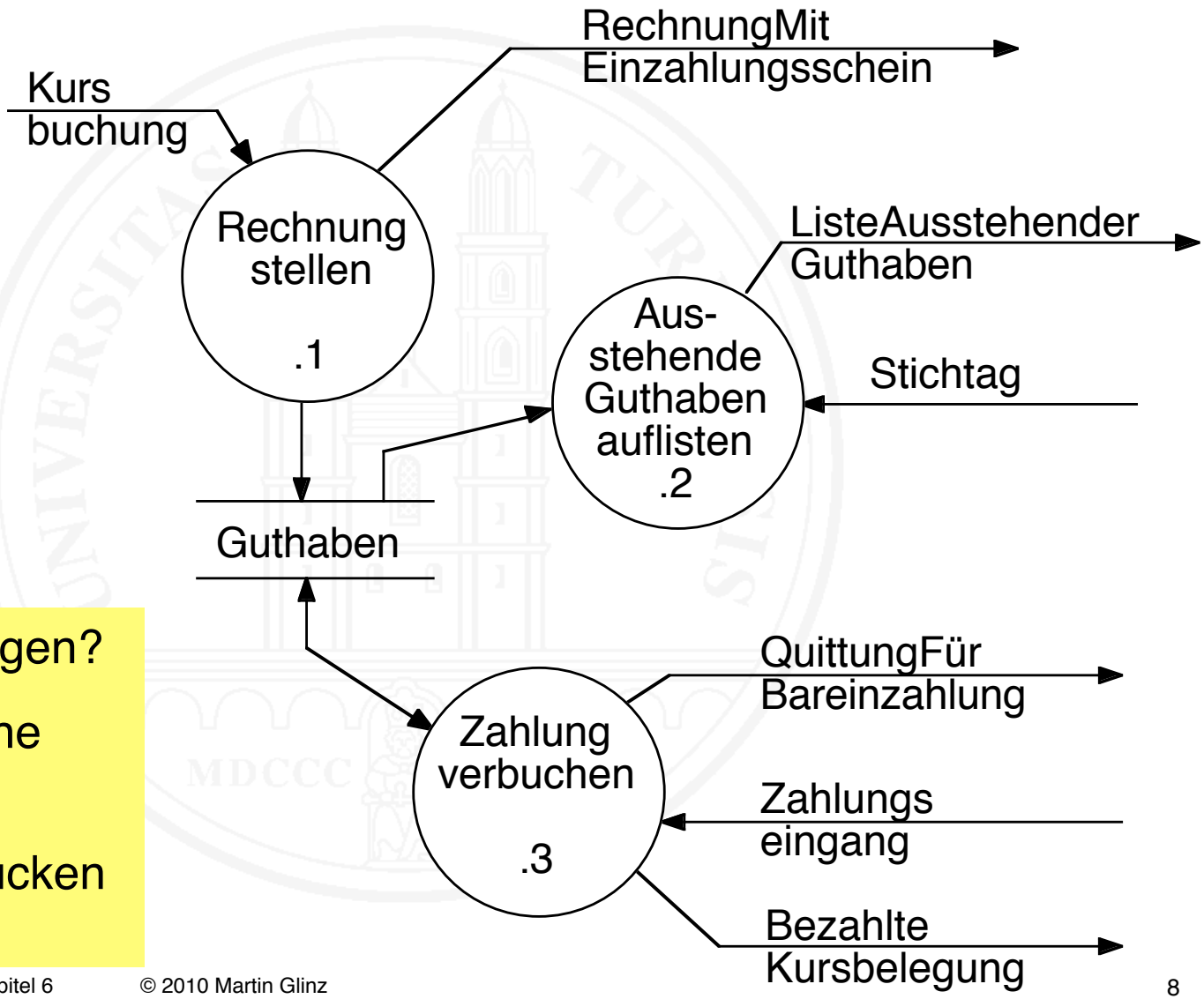


Aufgabe 6.1

Das nachstehende Datenflussdiagramm zeigt einen Ausschnitt aus dem Modell der Administration einer Volkshochschule.

Fassen Sie die Aussagen dieses Datenflussdiagramms in Worte. Wo können Sie (fast) sichere Aussagen machen, was ist nur wahrscheinlich, was wissen Sie nicht? Wie könnten Sie die Wissenslücken schließen?

Aufgabe 6.1 (Fortsetzung)



- Sichere Aussagen?
- Wahrscheinliche Aussagen?
- Wie Wissenslücken schließen?

Notwendige Ergänzungen

- Datenflussdiagramme sind Übersichtsmodelle. Zur Präzisierung müssen
  - die Namen aller Datenflüsse und Speicher definiert werden
  - die Funktionalität jeder Aktivität beschrieben werden

- Eine Aktivität kann wiederum durch ein Datenflussdiagramm beschrieben werden → DFD-Hierarchie
- Es braucht eine Notation zur Beschreibung elementarer, nicht zerlegter Aktivitäten

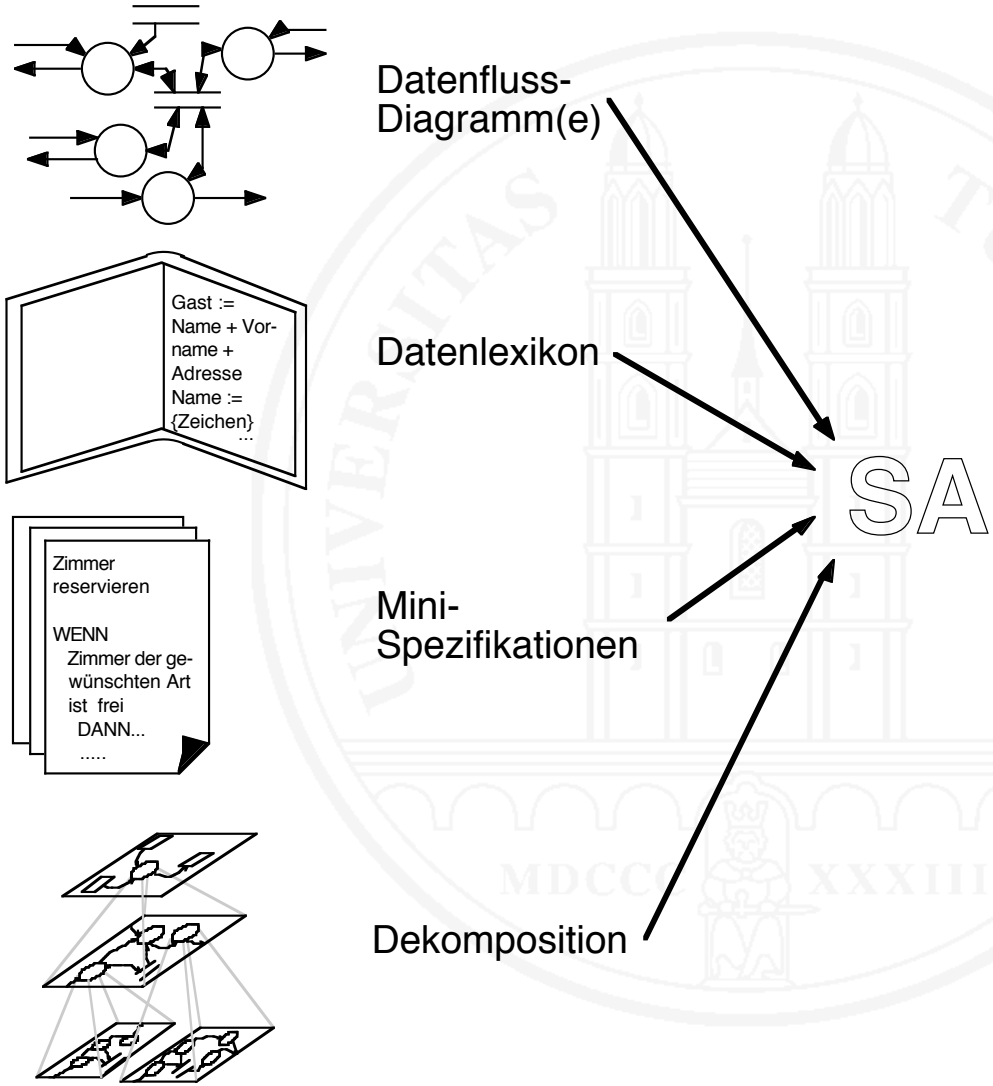
6.3 Strukturierte Analyse

Strukturierte Analyse (structured analysis) ist das klassische Beispiel einer datenflussorientierten Methode
(DeMarco 1978, Gane und Sarson 1979, McMenamin und Palmer 1984, Yourdon 1989)

SA verwendet vier Konzepte:

- Datenflussdiagramme als zentrales Modellierungsmittel
- Ein Datenlexikon zur Datendefinition
- Mini-Spezifikationen zur Beschreibung elementarer Aktivitäten
- Hierarchische Zerlegung der DFD zur Strukturierung des Modells

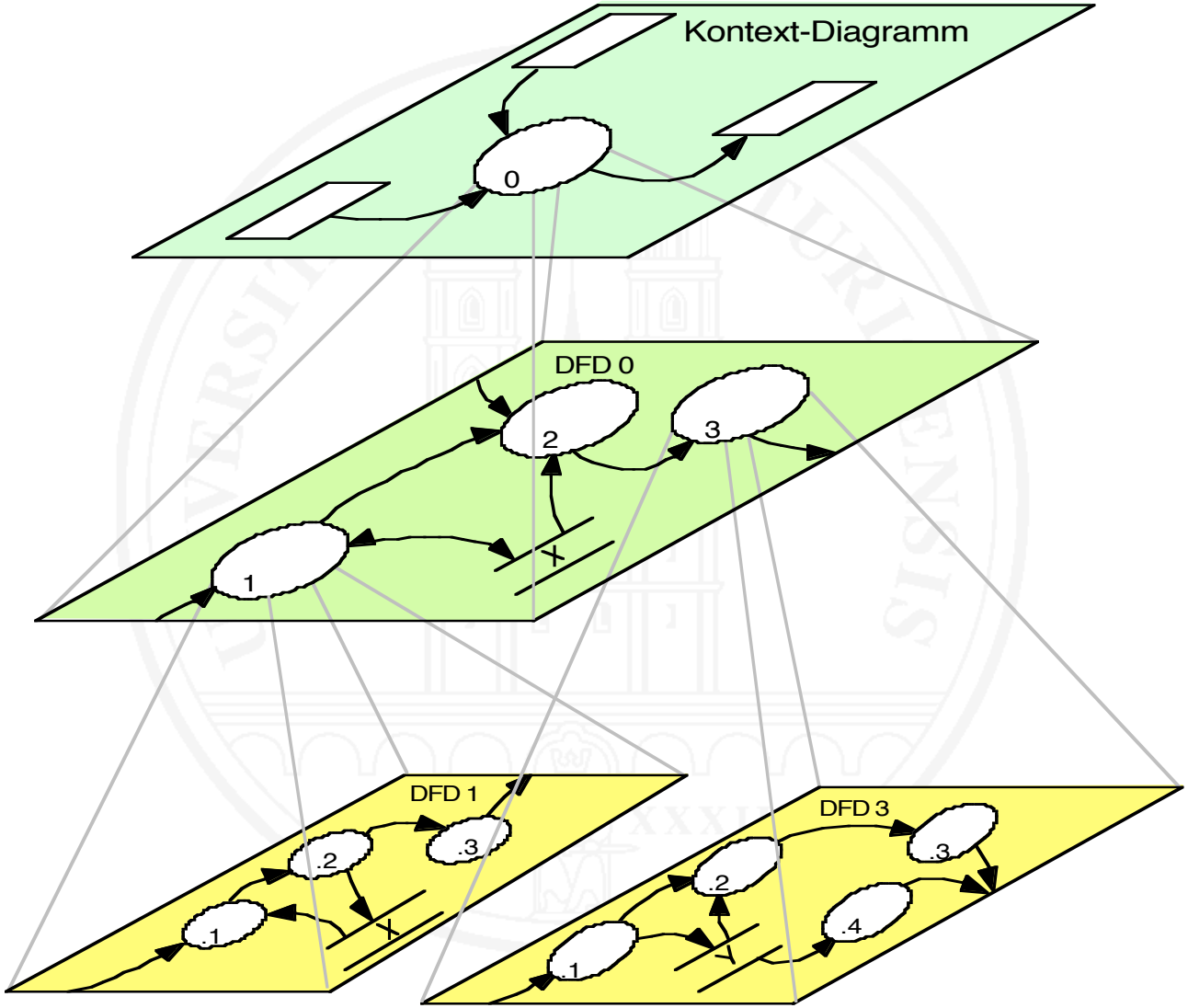
Grundkonzepte der Strukturierten Analyse



Hierarchische Zerlegung

- Die Datenflussdiagramme eines Systems können hierarchisch in Schichten angeordnet werden
- Jede Ebene fasst die Datenflussdiagramme der darunterliegenden Ebenen zu je einer Aktivität zusammen
- In der Regel wird ein dazu passendes hierarchisches Nummerierungs-schema für Aktivitäten und DFD verwendet
- Bei geeigneter Wahl der Zerlegung kann so ein komplexes Modell schrittweise in einfachere Teilmodelle zerlegt werden

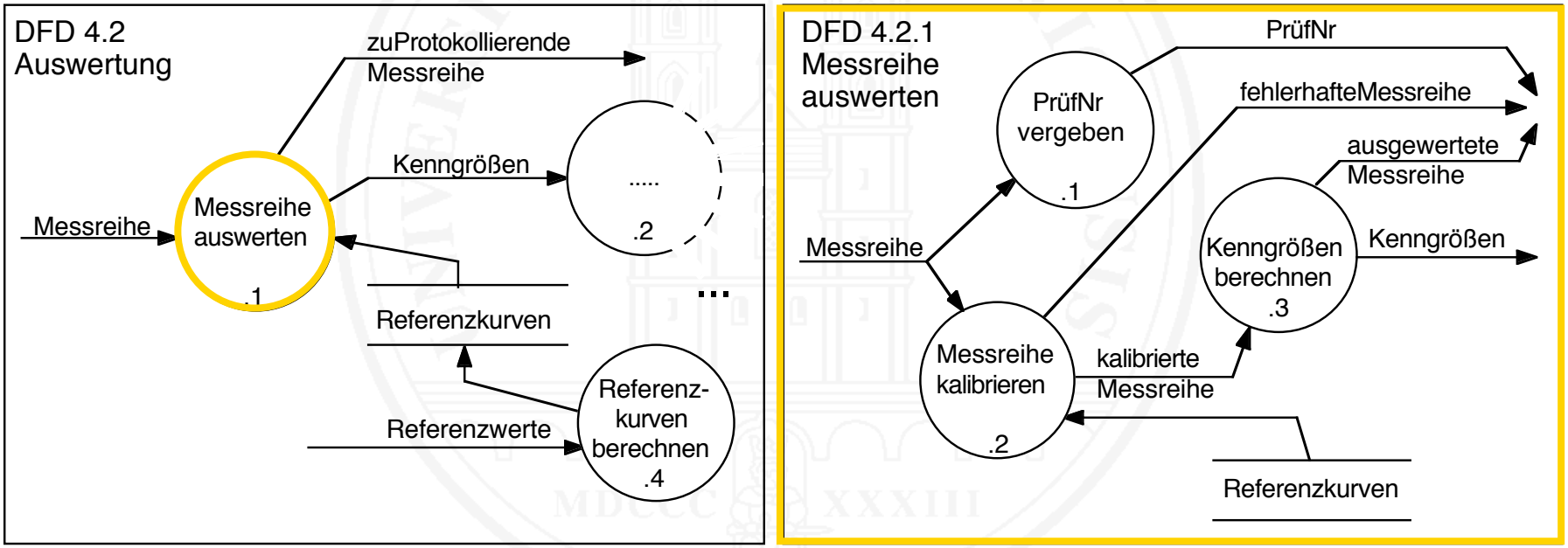
Beispiel einer DFD-Hierarchie in SA



Konsistenz der Hierarchieebenen

In jeder DFD-Hierarchie müssen die Diagramme auf den verschiedenen Ebenen miteinander konsistent sein.

Beispiel: zwei konsistente DFD:



Auszug aus dem Datenlexikon: zuProtokollierendeMessreihe := PrüfNr + [ausgewerteteMessreihe | fehlerhafteMessreihe]

Modelltheoretische Konzepte in SA

Modellelement	Modelltheoretisches Konzept
Aktivität, Speicher, Endknoten	Individuum
Datenfluss	Attribut
Datenlexikon-Eintrag	Attribut
Mini-Spezifikation	Attribut
Problembereich	Original
Abbildung der Flussstruktur	Abbildungsmerkmal
(beispielsweise) keine Beziehungen zwischen Speichern	Verkürzungsmerkmal
Auswahl und Granularität von Aktivitäten und Speichern	pragmatisches Merkmal
Kreis, Doppellinie, Pfeil ...	Notation

6.4 Methodik der Modellerstellung

Klassisches Vorgehen (event partitioning, McMenamin und Palmer 1984)

1. Erstelle ein Kontext-Diagramm (vgl. Kapitel Interaktionsmodelle)
2. Erstelle eine Ereignisliste (Ereignis = Anstoß aus der Außenwelt, der eine Reaktion des Systems erfordert)
3. Zeichne ein DFD mit einer Aktivität für jeden Vorgang (= Verarbeitung eines Ereignisses und Erzeugung der erwarteten Reaktionen)
   - Schließe interaktiv die dabei auftretenden Informationslücken
   - Definiere die verwendeten Daten
   - Skizziere Mini-Spezifikationen
4. Restrukturiere dieses (in der Regel sehr große) DFD in eine Hierarchie
5. Vervollständige das Modell

Methodik der Modellerstellung

Das klassische Vorgehen hat bei großen Modellen erhebliche Schwächen

➪ **Besseres Vorgehen:**

2' Gliedere die Aufgabe in Teile, die (soweit wie möglich) in sich geschlossen sind

- Erstelle die Ereignisliste
- Bilde Unter-Ereignislisten für jede Teilaufgabe

3' Arbeite gemäß 3 für jede Teilaufgabe; stimme dabei das Neue mit schon vorhandenen Teilmodellen ab

4' Arbeite gemäß 4 für jede Teilaufgabe (falls nötig)

5' Arbeite gemäß 5 und synthetisiere die oberste(n) Zerlegungsebene(n) aus den Teilmodellen

Aufgabe 6.2

Ein Kurs-Informationssystem für eine Volkshochschule ist zu spezifizieren.

Folgende Informationen sind Ihnen bekannt:

Meldet sich jemand zu einem Kurs an, so werden der belegte Kurs sowie Name und Adresse der angemeldeten Person erfasst. Das Kursgeld kann entweder direkt bar bezahlt oder mit einem bei der Anmeldung erzeugten Einzahlungsschein eingezahlt werden. Eingehende Zahlungen werden verbucht; nur Bareinzahlungen werden quittiert. ...
Bis eine Woche vor Kursbeginn kann eine angemeldete Person sich wieder abmelden. Hat sie das Kursgeld bereits bezahlt, wird der Betrag von der Buchhaltung zurückerstattet. Die Abmeldung wird schriftlich bestätigt. ...
Eine Woche vor Kursbeginn wird automatisch eine Liste aller Kursteilnehmer erzeugt. ...

a) Erstellen Sie aufgrund dieser Informationen eine Ereignisliste.

Aufgabe 6.2 (Fortsetzung)

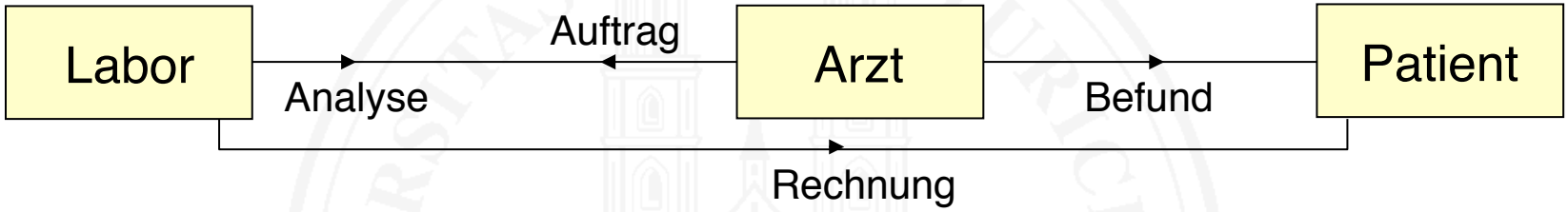
b) Modellieren Sie mit Hilfe der Ereignisliste das Problem der Kursadministration als Datenflussdiagramm. Stimmen Sie Ihr Modell mit dem Modell der Buchhaltung (Aufgabe 6.1) ab.

6.5 Weitere Ansätze

- SSADM (Structured Systems Analysis and Design Methodology)
  - eine auf Datenfluss basierende Methode zur Analyse und zum Entwurf von Systemen
  - Hauptelemente
    - Datenflussdiagramme
    - Datenmodell (basierend auf Entity-Relationship-Diagrammen)
    - Gegenstands-/Ereignismodell (von Jackson-Diagrammen abgeleitetes, spezielles Ablaufmodell)
  - In Großbritannien teilweise immer noch verbreitet
  - Sonst heute nicht mehr verwendet

Daten- bzw. Informationsfluss in UML

- UML 2 ermöglicht die Modellierung von Informationsfluss



- Vorgesehener Verwendungszweck:
  - Erstellung von Übersichtsmodellen: Systemelemente und deren Informationsaustausch
- Nicht vorgesehen zur Erstellung detaillierter, hierarchisch strukturierter Datenflussmodelle

Literatur

DeMarco, T. (1979). *Structured Analysis and System Specification*. New York: Yourdon Press.

Gane C., T. Sarson (1979). *Structured Systems Analysis: Tools and Techniques*. Englewood Cliffs, N.J.: Prentice Hall.

McMenamin, S.M., J.F. Palmer (1984). *Essential Systems Analysis*. New York: Yourdon Press.

Object Management Group (2015). *OMG Unified Modeling Language (OMG UML)*, version 2.5. OMG document formal/2015-03-01. http://www.omg.org/spec/UML/2.5

Yourdon. E.N., L.L. Constantine (1978). *Structured Design*. Englewood Cliffs, N.J.: Prentice Hall.

Yourdon, E. (1989). *Modern Structured Analysis*. Englewood Cliffs, N.J.: Prentice Hall.

Weaver, P.L. (1998). *Practical SSADM Version 4: A Complete Tutorial Guide*. Second Edition. Trans-Atlantic Publications.